AWS Resource usage Tracking Script

1. Introduction

This document provides details on how to set up a shell script that reports AWS resource usage in your project. Monitoring AWS services ensures cost efficiency by tracking whether all created resources are actively used.

2. Why Monitor AWS Usage?

AWS follows a pay-as-you-go model, meaning costs are directly tied to usage. To optimize expenses, tracking resource usage is crucial.

Key benefits include:

- • Management: Helps in understanding active and unused resources.
- • Cost Optimization: Ensures that unnecessary resources are identified and terminated.
- • Automated Reporting: Using a scheduled cron job, a daily usage report is generated.

3. AWS Resources to Monitor

This script tracks usage of the following AWS services:

- • S3: Lists all available S3 buckets.
- • EC2: Lists all active EC2 instances.
- • Lambda: Lists all deployed AWS Lambda functions.
- • IAM Users: Lists all AWS IAM users.

4. Prerequisites

Ensure the following prerequisites are met before running the script:

- • AWS CLI installed (`aws configure` should be set up with valid credentials).
- • Proper IAM permissions to list resources.
- • Bash shell available on the system.

5. Shell Script Implementation

Below is the shell script that retrieves AWS resource usage data and stores it in a file called resourceTracker.

```
#!/bin/bash
set -x
OUTPUT_FILE='resourceTracker'

echo 'Listing S3 Buckets...' | tee -a $OUTPUT_FILE
aws s3 ls >> $OUTPUT_FILE
```

```
echo 'Listing EC2 Instances...' | tee -a $OUTPUT_FILE
aws ec2 describe-instances --query 'Reservations[*].Instances[*].InstanceId' >>
$OUTPUT_FILE

echo 'Listing Lambda Functions...' | tee -a $OUTPUT_FILE
aws lambda list-functions >> $OUTPUT_FILE

echo 'Listing IAM Users...' | tee -a $OUTPUT_FILE
aws iam list-users >> $OUTPUT_FILE

echo 'AWS Resource Tracking Completed. Report saved in $OUTPUT_FILE.'
```

6. Running the Script

1. Make the script executable:

- chmod +x aws_resource_tracker.sh

2. Execute the script manually:

- ./aws_resource_tracker.sh

3. Verify the output:

- cat resourceTracker

7. Automating with Cron Job

To run this script daily at 6 AM, set up a cron job:

- • Open crontab: `crontab -e`
- • Add the following entry: `0 6 * * * /path/to/aws_resource_tracker.sh`

8. Expected Output Format

Sample output stored in resourceTracker file:

```
Listing S3 Buckets...
2025-01-25 10:35:46 my-s3-bucket

Listing EC2 Instances...
"i-1234567890abcdef0"

Listing Lambda Functions...
{
  'Functions': []
}

Listing IAM Users...
```

```
{
  'Users': [
    {
      'UserName': 'keerthi',
      'UserId': 'AIDAVVZPCQNSF5N2JHQHX'
    }
  ]
}
```

9. Troubleshooting

- • AWS CLI Not Configured: Run `aws configure`
- • Missing Permissions: Ensure IAM policies include `AmazonS3ReadOnlyAccess`, `AmazonEC2ReadOnlyAccess`, `AWSLambdaReadOnlyAccess`, and `IAMReadOnlyAccess`.
- • Cron Job Not Running: Check scheduled jobs with `crontab -l`.

10. Conclusion

This shell script automates AWS resource usage tracking and ensures cost efficiency. By running it daily via cron jobs, project teams can keep track of their AWS consumption and optimize usage. ▯